**Estado da Arte**

Esta secção tem como propósito apresentar um breve estudo do estado-da-arte de métodos de planeamento de trajetória e movimento de *robots* que se movimentam numa superfície plana com obstáculos.

De acordo com o artigo **[1]***“Path Planning for the Mobile Robot: A Review”*liderado por *Han-ye Zhang*, a aplicação adequada de tecnologias e metodologias de **planeamento de trajetos** adaptada a *robots* móveis contribui, não só para uma redução significativa do **tempo de execução** das tarefas previamente definidas, mas também reduzem o **desgaste** e o **investimento financeiro** no desenvolvimento destas soluções autónomas.

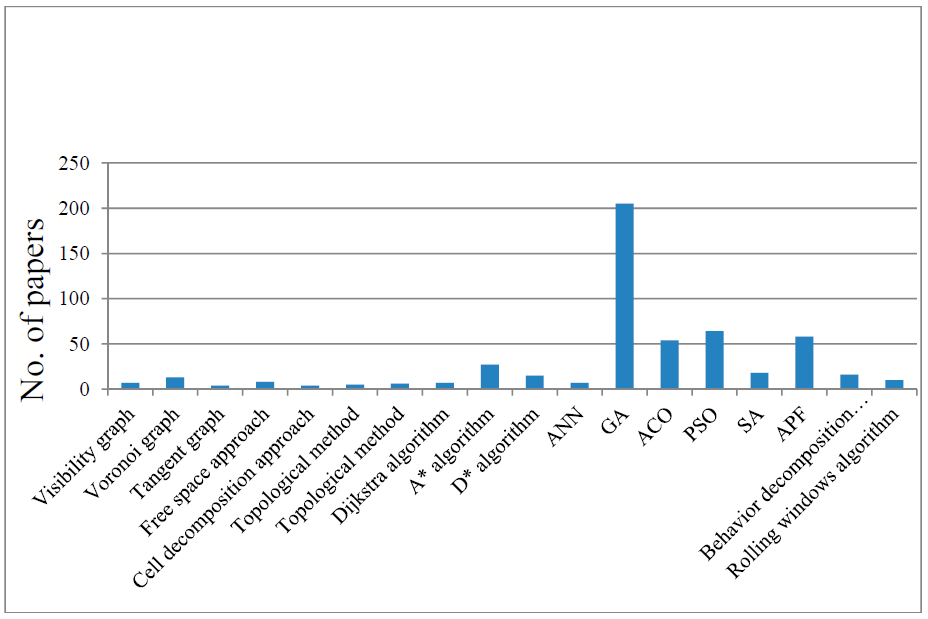
Várias propostas tecnológicas para melhorar o processo e os resultados do planeamento de movimento de *robots* móveis são propostas na literatura científica atual.

Embora estas técnicas não garantam uma solução **ótima global**, têm sido aplicadas em alguns trabalhos com relativo sucesso, contribuindo para a modelagem, optimização e implementação e execução algorítmica de movimentos destes *robots*.

As técnicas algorítmicas que mais se destacam para esta finalidade são:

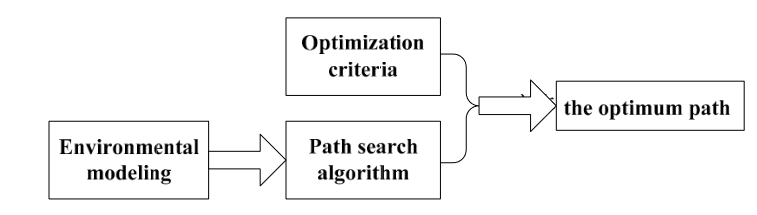
* GA (*genetic algorithm*)
* PSO (*particle swarm optimization algorithm*)
* APF (*artificial potential field*)
* ACO (*ant colony optimization algorithm*)

Como se pode observar no gráfico de barras da **Figura 1.** extraido do artido referenciado em [1] que tem como objetivo ilustrar o número de utilizações de cada técnica referida acima, podemos concluir que a implementação de soluções baseadas em algoritmos genéticos é a metodologia **predominante** para abordar problemas de planeamento.



**Figura 1.[1]** – Número de artigos publicados na base de dados da Engineering Village

É ainda referido no artigo[1] que o processo de determinação de trajetória de *robots* móveis segue normalmente os passos referido na **Figura 2.**



**Figura 2.[1]** - Fases fundamentais do processo de planeamento de caminhos

Estas fases são, em primeiro lugar, a **modelação do ambiente** em que o *robot* está inserido.

Para isto efeito é realizado um *maping* do ambiente em seu redor, obtendo para isso informação acerca dos potenciais obstáculos nas suas próximidades.

Depois de obter esta informação atravéz de sensores ou outros meios, codifica esta informação numa estrutura lógica, por exemplo numa matriz ou outra estrutura mais avançada.

Em seguida são aplicados **crtiérios de optimização**, com base em heurísticas, entre outras técnicas que complementando os algoritmos de pesquisa resultam num **caminho ótimo local** que pode ou não ser um ótimo global.

Os principais métodos para codificar o meio envolvente são, segundo o artigo[1], os seguintes:

* *Framework Space Aproach*
* *Free Space Approach*
* *Cell Decomposition Approach*
* *Topological Method*
* *Probabilistic Roadmap Method*

*Framework Space Aproach*

Com o objetivo de reduzir a complexidade da representação, o *robot* móvel é normalmente reduzido a um ponto e os obstáculos à volta do *robot* são reduzidos à escala, podendo o *robot* mover-se livremente no espaço dos obstáculos sem colidir com obstáculos e limites. Esta abordagem inclui o *visibility graph*, *voronoi graph*, *tangent graph*, sendo estas diferentes formas de repesentar os obstáculos.

Cada uma destas técnicas apresenta pontos fortes e menos fortes, por exemplo a nível de **rapidez de construção** ou **utlização de espaço em memória**.

*Free Space Approach*

Esta técnica, baseado no conceito de *free link* e descrita no artigo[2] refere uma nova técnica para **representar o espaço livre** obtível entre obstáculos em termos de uma região convexa livre.

É criado um novo grafo chamado *MAKLINK* que será utilizado para gerar um caminho que contorna os obstáculos.

Este grafo é construído através do uso dos pontos médios de ligações livres entre a região convexa livre como pontos de passagem. Estes pontos representam nós, e a conexão entre os pontos dentro de cada região convexa representa arcos no grafo.

A vantagem desta abordagem é que é mais flexível e pode alterar facilmente o ponto de partida e o ponto de destino do *robot*. Contudo, num espaço densamente populado com obstáculos, a abordagem de *free link* pode falhar e não conseguir obter um caminho ótimo.

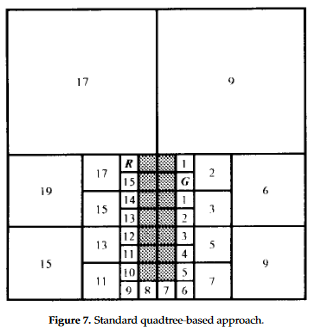
*Cell Decomposition Approach*

Este método decompõe o espaço de movimento do *robot* em regioes simples, chamadas neste contexto de células.

Estas celulas formam um grafo conexo e é procurado um caminho desde a celula inicial até à célula destino da segunte forma:

* Divisão sucessiva das regiões em células cada vez menores
* Se alguma das celulas contiver um obstáculo, esta é dividida em 4 novas células
* O processo de divisão celular é repetido até que se obtenha a localiza precisa dos obstáculos
* Aplicação do algoritmo de pesquisa de caminho

Podemos visualizar na **Figura 3.**[1] as divisões sucessivas que foram sendo realizadas até serem encontrados os obstáculos, para posterior aplicação do algoritmo de pesquisa.



**Figura 3[1].** Divisão sucessiva do espaço de pesquisa

**Algoritmos de pesquisa de caminho**

Os métodos de pesquisa de caminhos podem ser classificados em dois tipos:

* Baseados em Heurísticas
* Baseados em Inteligencia Artificial

Os métodos baseados em heurísticas mais utilizados são os seguintes:

* *Algoritmo de Dijkstra*

Proposto por *E.W Dijkstra* em 1959 tem como objetivo encontrar o caminho mais curto num grafo direcionado.

Tem como cacterística fundamental o facto de o ponto de partida ser o centro a partir do qual vai ser encontrado o caminho até ao ponto de destino. Em cada iteração o algoritmo seleciona o vértice que minimize o somatório do peso dos ramos desde o vértice de partida até esse mesmo vértice.

Não é um algoritmo muito eficiente pois pode indicar um caminho que atrevesse mais nós do que necessário.

* *Algoritmo A\**

Proposto por *Hart* em 1968 tem por base o algoritmo de *Dijkstra*.

A partir de um determinado nó específico, o valor pesado dos nós filhos actuais é atualizado e o nó filho que tiver o valor pesado mais pequeno é utilizado para atualizar o nó atual até que todos os nós sejam percorridos. A chave do algoritmo A\* é estabelecer a função de avaliação f (n), f (n) = g(n) + h(n), em que g(n) representa o custo real do nó inicial para o nó n, e h(n) representa o custo estimado do caminho ótimo do nó n para o nó de destino no espaço de estados.

* Algoritmo D\*

Stentz propôs o D\*em 1994.

É utilizado principalmente pelos *robots* para deteminar um caminho.

O espaço problema do algoritmo D\* é expresso como uma serie de estados que representam

a direcção da posição do *robot*.

O princípio base do algoritmo D\*é semelhante ao do A\*onde o peso do ramo é usado para definir a direcção da pesquisa.

Os métodos baseados em inteligência artificial mais utilizados são os seguintes:

* ANN (*Artificial Neural Networks*)
* GA (*Genetic Algorithms*)
* ACO (*Ant Colony Optimization*)
* PSO (*Particle swarm optimization*)

*Artificial Neural Networks*

As redes neuronais são utilizadas para descrever os obstáculos dentro do ambiente e a energia é definida em função do ponto do *path*. O nivel de energia depende da localização do ponto do *path*, movendo-se o *robot* na direção da menor energia. Um *path* com a menor energia total é obtida no final.

O uso de redes neuronais não é muito utilizado pois é dificil definir a topologia de uma rede que descreva um ambiente em movimento.

*Particle Swan Optimization*

Inspirados pela consistência da atividade de grupos de aves, *Eberhart e Kennedy* propuseram o algoritmo PSO (otimização por enxame de partículas) em 1995.

O algoritmo parte de uma solução aleatória. Encontra a solução óptima através de sucessivas iterações. Avalia a qualidade da solução através do valor de aptidão, e encontra o ótimo global, comparando o valor ótimo atualmente procurado com o último.

Este algoritmo é utilizado para resolver o planeamento de trajectórias robóticas com as vantagens de fácil implementação, elevada precisão e rápida convergência.

**Referências**

[1]Han-ye Zhang; Wei-ming Lin; Ai-xia Chen, “Path Planning for the Mobile Robot: A Review”, 2018, School of Mechanical & Materials Engineering, Jiujiang University, China

[2]Habib, M.K.; Asama, H., Efficient method to generate collision free paths for an autonomous mobile robot based on new free space structuring approach., 1991, Osaka, Japan